# Compression of ECG Signal Using Neural Network Predictor and Huffman Coding

Ridha Iskandar
Gunadarma University
Jl. Margonda Raya No. 100 Depok
Jawa Barat Indonesia
ridha@staff.gunadarma.ac.id

I Wayan Simri W
Gunadarma University
Jl. Margonda Raya No. 100 Depok
Jawa Barat Indonesia
iwayan@staff.gunadarma.ac.id

**Abstract**

**Medical signals and images need special treatment especially when the data become bigger and bigger. One of the treatment that will be considered in this experiment is about the data compression. ECG (Electro Cardio Graph) signals will create very big data when the signals were collected in a long period of time. Several methods can be used to compress the ECG data. In this experiment we used neural network to predict the incoming data and huffman coding to minimize the codes. The ECG data was collected from MIT-BIH arrhythmia database. The experiment gave low compression rasio when the predicted data was very close to the incoming data.**

***Keywords-component; compression, ECG, huffman code, neural network***

## I. INTRODUCTION

Human heart has important role in the human body. It pumps blood through the whole body over 100,000 times daily [1]. The ECG signal which represents heart activity has a graphical form in Figure 1. The P wave marks the activation of the atria, which are the chambers of the heart that receive blood from the body. The activation of the left atrium, which collects oxygen-rich blood from the lungs, and the right atrium, which gathers oxygen-deficient blood from the body, takes about 90

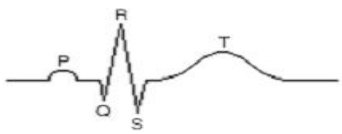


Figure 1. Important feature of ECG signal [1].

msec. Next in the ECG cycle comes the QRS complex. The heart beat cycle is measured as the time between the second of the three parts of the QRS complex, the large R peak. The QRS complex represents the activation of the left ventricle, which sends oxygen-rich blood to the body, and the right ventricle, which sends oxygen-deficient blood to the lungs. During the QRS complex, which lasts about 80 msec, the atria prepare for the next beat, and the ventricles relax in the long T wave. It is these features of the ECG signal by which a cardiologist uses to analyze the health of the heart and note various disorders, such as atrial flutter, fibrillation, and bundle branch blocks [1].

An ECG signal, according to the American Heart Association, must consist of 3 individual leads, each recording 10 bits per sample, and 500 samples per second. Some ECG signals, may require 12 leads, 11 bits per second, 1000 samples per second, and last 24 hours. When converted to a digital format, this single ECG record requires a total of 1.36 gigabytes of computer storage! Considering the 10 million ECGs annually recorded for the purposes of comparison and analysis in the United States alone, the necessity for effective ECG data compression techniques is becoming increasingly important. Further more, the growing need for transmission of ECGs for remote analysis is hindered by capacity of the average analog telephone line and mobile radio [1].

Data compression techniques are categorized as those in which the compressed data is reconstructed to form the original signal and techniques in which higher compression ratios can be achieved by introducing some error in the reconstructed signal. These kinds of techniques are called lossy. The techniques which introduce zero error in the reconstructed signal are called lossless. The effectiveness of an ECG compression technique is described in terms of compression ratio (CR), a ratio of the size of the compressed data to the original data; execution time, the computer processing time required for compression and reconstruction of ECG data; and a measure of error loss, often measured as the percent mean-square difference (PRD). The PRD is calculated as follows:

$$PRD = 100 \cdot \sqrt{\frac{\sum_{i=1}^{n}(ORG(i) - REC(i))^2}{\sum_{i=1}^{n}(ORG(i))^2}}$$

Data compression techniques are categorized as those in which the compressed data is reconstructed to form the original signal and techniques in which higher compression ratios can be achieved by introducing some error in the reconstructed signal. These kinds of techniques are called lossy. The techniques which introduce zero error in the reconstructed signal are called lossless. The effectiveness of an ECG compression technique is described in terms of compression ratio (CR), a ratio of the size of the compressed data to the original data; execution time, the computer processing time required for compression and reconstruction of ECG data; and a measure of error loss, often measured as the percent mean-square difference (PRD). The PRD is calculated as follows

where ORG is the original signal and REC is the reconstructed signal. The lower the PRD, the closer the reconstructed signal is to the original ECG data [1].

Lossless compression schemes are preferable to lossy compression schemes in biomedical applications where even the slight distortion of the signal may result in erroneous diagnosis [2]. There are many kinds of lossless techniques, one of which is statistical encoding. In statistical encoding, data samples are replaced by specially chosen codes. The frequencies and probabilities of occurrence of single samples are used to create codes in which short codes represent frequently occurring samples, while longer codes represent rare occurrences. One of the first statistical encoding techniques is the familiar Morse code, in which a single dot or dash replaces the most common English letters *E* and *T*. Huffman coding is one of the most used statistical compression methods since it contains the shortest average code length. In Huffman coding, the most frequently occurring data sample is replaced by a simple binary code. Each following data sample, in the order of occurrence, is replaced by a different code, equal to or greater in length than the previous code. Huffman coding assumes that the average code length will be shorter than the average original data sample. In ECG data compression, Huffman coding is often used as a final compression method, after compression techniques that introduce error have been applied since Huffman coding results in no further error [1].

There is a method to do compression by doing prediction of the incoming data. This method actually uses a compression technique that introduces some error. Polynomial predictors including zero-order and first-order predictors, polynomial interpolators such as the zero-order and first-order interpolators, AZTEC (amplitude zone time epoch coding), TP (turning point), CORTES (coordinate reduction time encoding system), and average beat subtraction techniques are commonly applied to ECG records. More complex techniques such as the use of wavelet packets [3], neural networks, and adaptive Fourier coefficients are currently being explored with the expectations that they will result in higher compression ratios but longer processing time for compression and reconstruction [1]. Some techniques also consider the periodicities of the signals like we see in [4], [5] and [6].

By combining the method of prediction of the incoming data and the statistical method we can get another way of compressing data. For ECG signal compression we can use neural network as a predictor and Huffman coding as an entropy encoder to compress the data.

## II. LOSSLESS DATA COMPRESSION METHOD

The lossless data compression method can be seen in the Figure 2. Part of the input data is used to predict the incoming data by neural network predictor. The residues between the incoming data and the predicted one are used by the Huffman Code Generator to produce the binary code. The predicted figures and the binary codes are sent to the Signal Reconstruction to produce the output at the receiving end.

The predictor that is implemented at the method above is Back-Propagated Delta Rule networks. Back-Propagated Delta Rule Networks (BP) (sometimes known and multi-layer perceptrons (MLPs)) and Radial Basis Function Networks (RBF) are both well-known developments of the Delta rule for single layer networks (itself a development of the Perceptron Learning Rule). Both can learn arbitrary mappings or classifications. Further, the inputs (and outputs) can have real values. Back-Propagated Delta Rule Networks (BP) is a development from the simple Delta rule in which extra hidden layers (layers additional to the input and output layers, not connected externally) are added. The network topology is constrained to be *feedforward*: i.e. loop-free - generally connections are allowed from the input layer to the first (and possibly only) hidden layer; from the first hidden layer to the second,.and from the last hidden layer to the output layer.

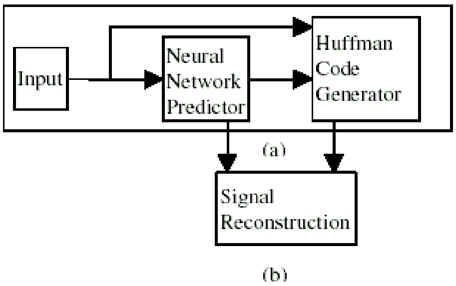


Figure 2. Lossless compression Method (a) transmitting end (b) receiving end.

The hidden layer learns to *recode* (or to *provide a representation* for) the inputs. More than one hidden layer can be used. The architecture is more powerful than single-layer networks: it can be shown that any mapping can be learned, given two hidden layers (of units). The units are a little more complex than those in the original perception:

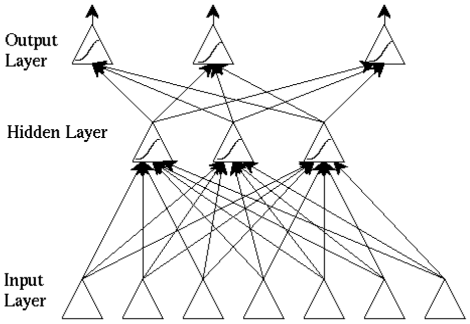


Figure 3. Typical BP network architecture.

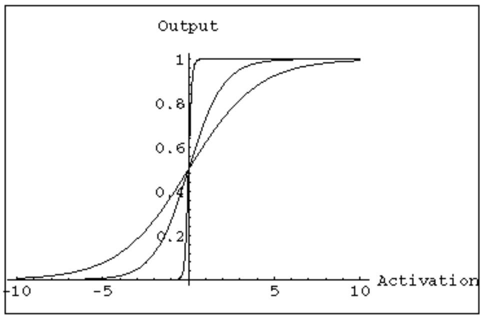


Figure 4. The input/output graph of BP network.

their input/output graph is shown Figure 4. As a function:

$Y = 1 / (1+ \exp(-k.(\Sigma W_{in} * X_{in}))$

The graph shows the output for *k=0.5, 1, and 10,* as the activation varies from -10 to 10.

Huffman Code assigns shorter encodings to elements with a high frequency. It differs from block encoding in that it is able to assign codes of different bit lengths to different elements. Elements with the highest frequency get assigned the shortest bit length code. The key to decompressing huffman code is a huffman tree.

A huffman tree is a special binary tree called a trie. (pronounced try). A binary trie is a binary tree in which a 0 represents a left branch and a 1 represents a right branch. The numbers on the nodes of the binary trie represent the total frequency of the tree. The leaves of the trie represent the elements to be encoded. The elements are assigned the encoding which corresponds to their place in the binary trie.

## III. Result and Discussion

There are several steps to do the compression. The first step is to give part of the input to the Neural Network Predictor so that the predictor can built the weights. These weights will be used by the neural network to predict the incoming input.

When the program is run the neural network will start training its neuron until the goal is reached. The input and the target output should be given to the network. The performance of the neural network can be seen in the Figure 5.

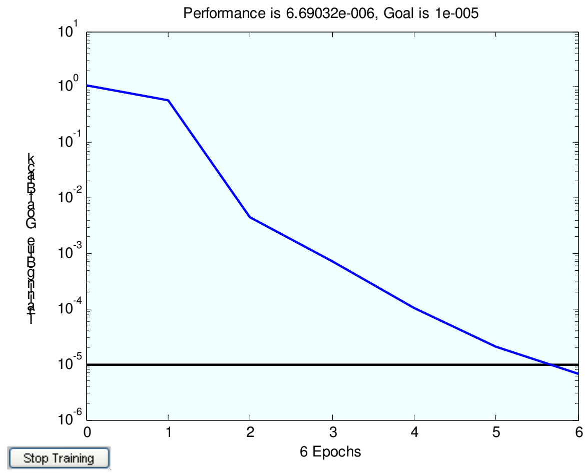


Figure 5. The Performance of neural network.

After the goal is reached the neural network will give the predicted data. The substraction between the incoming data and the predicted data will be converted into Huffman code. To create such kind of code the substraction should be put into integer number of the form uint8. By using uint8 format the data should be in the range between 0 and 256. The Huffman Code will assign shorter encodings to elements high frequency of occurence.

The residues between the predicted figures and the incoming input that have been converted into binary codes through Huffman Code Generator, and the predicted input will be given to the Signal Reconstruction via communication link. The signals will be reconstructed again.

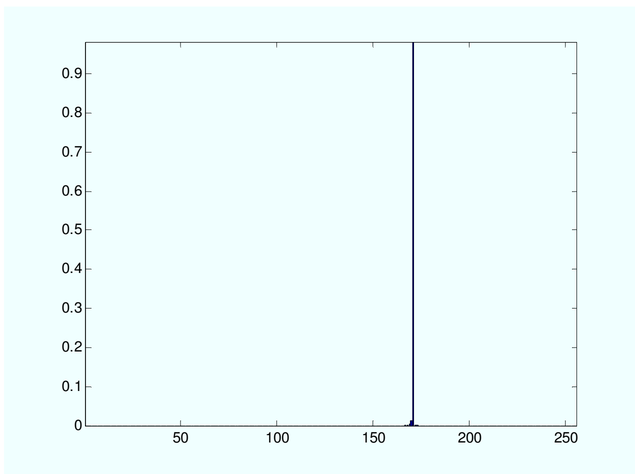


Figure 6. The result if the predicted input is the neural network product of the input itself.

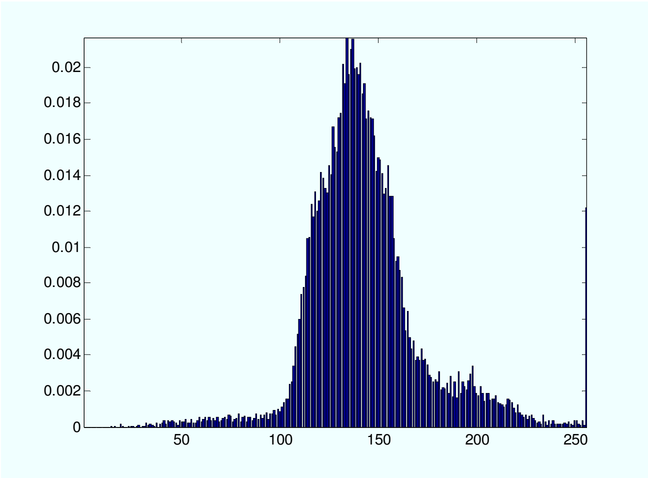


Figure 7. The result if the predicted input is not the neural network product of the input itself.

The frequency distribution of the data can be seen in the Figure 6. It shows the result of the frequency distribution of the substracted data when the incoming input is substracted to the predicted input. The predicted input is the neural network product from the input itself.

Figure 7 shows the result of the frequency distribution of the substracted data when the predicted input is the product of the neural network with different incoming input which part of them was used as a sample data.

In the experiment we used the data which is available at MIT-BIH arrhythmia database. When the incoming data is the same with the sample data we got the compression ratio of 0.1282. If we used other data we got compression rasio of 0.8128, 0.8519, 0.8067, 0.7278, 0.1361, and 0.6296. The low compression ratios are acquired when the predicted signals are close to the incoming signals.

## IV. Conclusion

The experiment of compressing ecg signals by using neural network as a predictor and huffman code will give low compression ratio if the prediction of the signals is close to the incoming signals. Following this result we will elaborate the prediction methods in order to get the one who give the best prediction.

## Acknowledgment

Our thanks to Professor Michel Paindavoine from Universite de Bourgogne France for his guidance during the research. Special thanks also to our colleagues in Gunadarma University in Indonesia for their supporting to do this research.

## References

[1] Anonymous, “A Comparison of Single Lead ECG Data Compression Techniques”.

[2] Kannan, R. and C. Eswaran, “Lossless Compression Schemes for ECG Signals Using Neural Network Predictors”, EURASIP Journal on Advances in Signal Processing, Volume 2007.

[3] Bilgin, A, Michael W. Marcellin, and Maria I. Altbach, “Wavelet Compression of ECG Signals by JPEG2000”, Proceedings of the IEEE Data Compression Conference, 2004.

[4] Brito, M, J. Henriques, P. Carvalho, B. Ribeiro, and M. Antunes, “An ECG Compression Approach Based on A Segment Dictionary and Bezier Approximations”, EURASIP, 2007.

[5] Schimming, T., M.Ogorzalek, H.Dedieu, “A Nonlinear Dynamical Model for Compression and Detection of ECG Data”, 2007.

[6] Sameni, R., Christian Jutten, and Mohammad B. Shamsollahi, “Multichannel Electrocardiogram Decomposition Using Periodic Component Analysis”, IEEE Transaction on Biomedical Engineering, Vol. 55, No. 8, August 2008.